Activation-Wise Propagation: A One-Timestep Strategy for Spiking Neural Networks

Jian Song[1], **Xiangfei Yang**[2*], **Shangke Lyu**[3], **Donglin Wang**[1*]
[1]Westlake University
[2]Hangzhou City University
[3]Nanjing University
songjian76@westlake.edu.cn, yxf9011@163.com, shangke_lyu@nju.edu.cn, wangdonglin@westlake.edu.cn

Abstract

Spiking neural networks (SNNs) have demonstrated significant potential in real-time multi-sensor perception tasks due to their event-driven and parameter-efficient characteristics. A key challenge is the timestep-wise iterative update of neuronal hidden states (membrane potentials), which complicates the trade-off between accuracy and latency. SNNs tend to achieve better performance with longer timesteps, inevitably resulting in higher computational overhead and latency compared to artificial neural networks (ANNs). Moreover, many recent advances in SNNs rely on architecture-specific optimizations, which, while effective with fewer timesteps, often limit generalizability and scalability across modalities and models. To address these limitations, we propose Activation-wise Membrane Potential Propagation (AMP2), a unified hidden state update mechanism for SNNs. Inspired by the spatial propagation of membrane potentials in biological neurons, AMP2 enables dynamic transmission of membrane potentials among spatially adjacent neurons, facilitating spatiotemporal integration and cooperative dynamics of hidden states, thereby improving efficiency and accuracy while reducing reliance on extended temporal updates. This simple yet effective strategy significantly enhances SNN performance across various architectures, including MLPs and CNNs for point cloud and event-based data. Furthermore, ablation studies integrating AMP2 into Transformer-based SNNs for classification tasks demonstrate its potential as a general-purpose and efficient solution for spiking neural networks.

Code — https://github.com/LockFaiz/AMP2
Extended version — https://arxiv.org/abs/2502.12791

Introduction

Due to the low power consumption and the emergence of neuromorphic sensors, Spiking Neural Networks (SNNs) have gained attention to address perception tasks. Unlike artificial nonlinear activation functions such as ReLU or sigmoid, SNNs encode representations using discrete spikes, mimicking the behavior of biological neurons. Taking the Leaky Integrate-and-Fire (LIF) neuron as an example, the spiking process consists of three stages (**Equation 1**–**Equation 3**): accumulation , emission, and reset. A hidden state known as membrane potential (MP) is used to regulate both historical and current information by decaying past MP while integrating incoming inputs.

$$U_i(t) = H_i(t-1) + X_i(t) \tag{1}$$
$$S_i(t) = Hea.(U_i(t) - V_{th}) \tag{2}$$
$$H_i(t) = \beta U_i(t)(1 - S_i(t)) + V_{th} S_i(t) \tag{3}$$

Here, i and t denote the layer and timestep, $U_i(t)$ and $H_i(t)$ represent the MP of neurons before and after activation at timestep t. V_{th} represents the threshold for spike generation. $Hea.(\cdot)$ denotes the Heaviside step function defined as $Hea.(x) = 1$ if $x \geq 0$, and $Hea.(x) = 0$ otherwise. The leaky mechanism depends on the decay factor β, which attenuates the influence of historical information.

SNNs have increasingly matched artificial neural networks (ANNs) in performance and efficiency, particularly narrowing the gap in training efficiency for static images. The Spike-Driven Transformer (Yao et al. 2023, 2024a,b) continually optimizes the spiking mechanism within the transformer architecture, utilizing only 4 timesteps for static images and 16 timesteps for neuromorphic data captured by dynamic vision sensors (DVS). Spikformer (Zhou et al. 2023, 2024b) introduces spiking self-attention and a convolution-attention hybrid mechanism, outperforming other ResNet-based SNNs on both static and DVS images. Notably, QKFormer (Zhou et al. 2024a) trained with 4 timesteps, surpasses several ANNs on ImageNet-1k for the first time.

For neuromorphic data, a common pre-processing involves integrating an event episode into a number of frames, which are required to match the timesteps of SNN. To preserve fine-grained information, researchers typically integrate a larger number of frames (often 10 or 16). This requirement leads to longer timesteps for SNN when processing DVS data compared to static images, further reducing the training efficiency of SNNs. Alternatively, some studies have proposed treating events as 3D event clouds (Wang et al. 2019) to circumvent the integration step. Although SpikePoint (Ren et al. 2023b) performs well on multiple DVS benchmarks, it still relies on the 16-timestep training. Therefore, reducing the number of timesteps remains a key challenge for SNNs.

*Corresponding author.

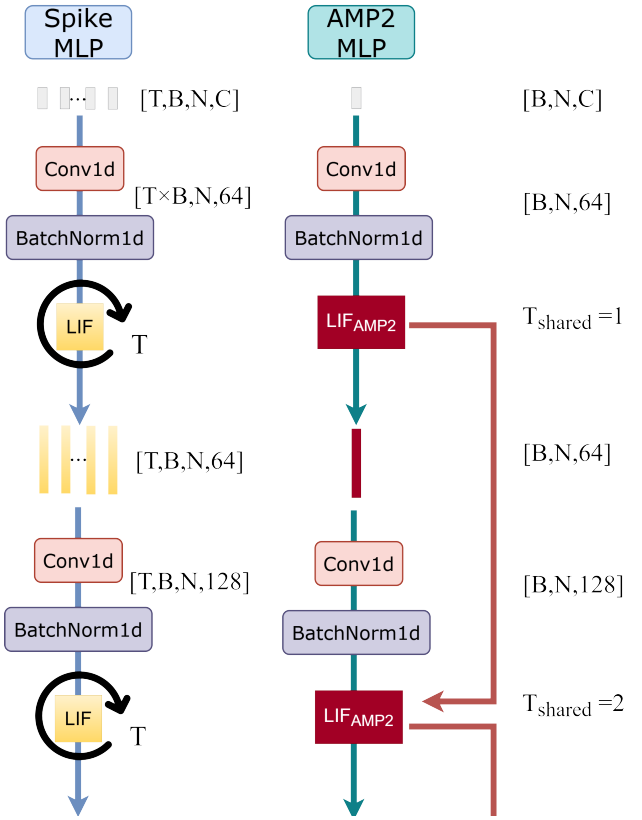

Figure 1: Forward propagation of Spiking PointNet. Left: **T**imestep-**W**ise **U**pdate (TWU). Right: **A**ctivation-wise **M**embrane **P**otential **P**ropagation (AMP2). AMP2 replaces timestep-wise iterative MP accumulation with neuron-wise accumulation, enabling an SNN that operates effectively at a single timestep.

Timestep	Memory per GPU	GPU Hours	Acc.(%)
1	30.15GB	42.96	78.0
4	74.87GB	98.64	80.36

Table 1: Training efficiency of Spikformer(8-512) on ImageNet100 with a batch size of 120 under different timesteps.

Such performance has not yet been achieved in the 3D point cloud domain. Wu et al. (2025) proposed Spiking Point Transformer (SPT), which exploited spiking self-attention for Point Transformer (Zhao et al. 2021). With increased model size, SPT readily outperforms the Spiking PointNet (Ren et al. 2023a). Similarly, Wu et al. (2024) scaled up convolutional SNN by stacking 13 spiking residual blocks in P2SResLNet, drawing upon KPConv (Thomas et al. 2019). Despite these advances in scaling, there remains a lack of **lightweight yet effective** methods for SNN on point clouds, whereas some techniques have demonstrated effectiveness for SNNs on 2D static images (discussed in **Section**).

The iterative **T**imestep-**W**ise **U**pdate (TWU, shown on the left of **Figure 1**) of neuronal MP remains the predominant approach in current SNN models. For static RGB images, these methods typically replicate the input along an additional temporal dimension to enable TWU. Specifically, an image input with shape $[B, C, H, W]$ is augmented with a temporal dimension T, resulting in input of shape $[T, B, C, H, W]$, where B, C, H, and W denote batch size, channels, height, and width, respectively. T determines the number of intermediate spike emissions, i.e., the number of iterative calculations for MP. A large T ensures stable spiking representations. However, this pre-processing introduces data redundancy, leading to excessive memory consumption and prolonged spiking iterations. **Table 1** shows that increasing timesteps from **1 to 4** results in a $\mathbf{2.5}\times$ increase in **memory** consumption and a $\mathbf{2.3}\times$ increase in training **time**.

When T is small, most spiking neurons may not reach the firing threshold, resulting in a narrow and underdeveloped MP distribution. Consequently, the generated spikes may inadequately represent the underlying information. Furthermore, TWU inevitably amplifies the variance of MP distribution during temporal iterations, regardless of the value of T, due to the variance accumulation of hidden states. To address the instability of SNN outputs, Ding et al. (2025) proposed MP smoothing and guidance, aiming to reduce inter-timestep variability in MP distribution. While this approach demonstrates effectiveness under moderate timesteps, it does not fundamentally resolve the issues of computational burden and latency caused by temporal iterations. This naturally raises the question of **whether it is possible to achieve high performance while eliminating timestep-wise iterative updates**, that is, an effective single-timestep MP update. Information accumulation through neighboring neurons in the human brain may be a solution.

Unlike conventional SNNs, where iterative MP updates and forward propagation alternate, we propose a parallel propagation scheme for both MP and spikes, eliminating the need for timestep-wise MP updates. This method, termed **A**ctivation-wise **M**embrane **P**otential **P**ropagation (AMP2) is illustrated in the right of **Figure 1**. The main contributions of this paper are summarized as follows:

- AMP2 consistently improves SNN performance without iterative updates ($T = 1$) across multiple architectures, including Spiking PointNet, Spiking ResNet, and Spiking Transformers.
- AMP2-based PointNet series outperform low-timestep baselines with more parameters on ModelNet40 (Wu et al. 2015) and ScanObjectNN(Uy et al. 2019).
- We conduct extensive experiments on multiple datasets, covering classification and semantic segmentation of point clouds, frame-based and point-based recognition on neuromorphic data, as well as static image classification on ImageNet100 (Ambityga 2023).

Related Works on Reducing Timesteps

Timestep reduction is an emerging area. Ding et al. (2024) proposed Shrinking SNN (SSNN), which divides SNN training into multiple stages with progressively decreasing timesteps. SSNN employs larger timesteps in the early stages and gradually reduces them in later stages. Each stage

involves a complete classifier training, ultimately achieving strong performance with average timesteps of 5 for DVS frames. Ding et al. (2025) identified excessive differences in MP distributions across timesteps as a significant factor disrupting SNN training. They demonstrate that smoothing and guidance on temporally adjacent MP effectively alleviate such differences at 5 timesteps. Zuo et al. (2024) introduced temporal reversal augmentation (TRR) to regularize SNN training. By reversing the temporal order of inputs or features, TRR enhances SNN robustness to varying timesteps. As a result, timesteps for neuromorphic and static datasets are further reduced to 5 and 2, respectively. Notably, TRR demonstrates compatibility with existing models including VGG-9, MS-ResNet34 (Hu et al. 2024), Spike-Driven Transformer (Yao et al. 2023), and PointNet series (Qi et al. 2017a,b). Spiking Point Transformer and P2SResLNet compensate for reduced timesteps by employing larger model architectures; specific parameter counts are provided in **Appendix**.

Method

Overall Framework

Activation-wise **M**embrane **P**otential **P**ropagation (AMP2) is explained based on the LIF neuron described in **Equation 1–Equation 3**. To provide an intuitive illustration of AMP2, we use simplified notations MP^0, MP^1, and MP^2 to represent AMP2's MP at the initialization, accumulation, and reset stages, respectively, distinguishing them from $H(t)$ and $U(t)$, which denote MP in accumulation and reset stages in conventional TWU neurons. We introduce two key novelties to the TWU of the LIF neuron, shown in **Figure 2** and pseudocode in **Appendix**.

- Activation/Neuron-**W**ise **P**ropagation (AWP): We explicitly enable direct propagation of MP between neurons, in parallel with the forward propagation of features. The initial MP^0 is primarily determined by the residual membrane potential MP^1 carried over from the previous spiking layer, rather than being initialized to a fixed value or random perturbation (Ren et al. 2023a).
- Residual Learning of MP (RMP): Residual learning helps mitigate spike degradation, especially in deep SNNs. Inspired by existing residual connection in SNNs, we introduce a novel residual pathway between MP and features.

Activation/Neuron-wise Propagation (AWP)

For a fair comparison, we adopt a surrogate gradient function for LIF neuron as used in Spiking PointNet (Ren et al. 2023a).

Forward Propagation:

$$MP_{random} \sim \mathcal{U}(0, c), \quad \text{where } c \in (0, 1] \tag{4}$$

$$MP_i^0 = \alpha MP_{i-1}^1 + (1-\alpha) MP_{random} \tag{5}$$

$$MP_i^1 = \beta MP_i^0 + X_i \tag{6}$$

$$S_i = Hea.(\frac{MP_i^1}{V_{th}} - c) \tag{7}$$

Surrogate Gradient Function:

$$x_i = \frac{MP_i^1}{V_{th}} \tag{8}$$

$$g(x_i) = \frac{tanh(k(x_i - c)) + tanh(kc)}{2tanh(kc)} \tag{9}$$

k represents the scaling factor that controls the $\tanh$ function; larger values of k make the surrogate gradient approach a hard boundary. c denotes the implicit threshold in the surrogate gradient function. The gradient of the loss with respect to the weights, $\frac{\partial \mathcal{L}_{CE}}{\partial W_i}$, is directly proportional to $\frac{\partial \mathcal{L}_{CE}}{\partial MP_i^0}$ (Ren et al. 2023a; Ding et al. 2025), which is given by

$$\frac{\partial \mathcal{L}_{CE}}{\partial MP_i^0} = \frac{\partial \mathcal{L}_{CE}}{\partial S_i} \cdot \frac{k \cdot \operatorname{sech}^2(k(x_i - c))}{2\tanh(kc)} \cdot \frac{\beta}{V_{th}} \tag{10}$$

According to **Equation 10**,we can find that k, c, β, V_{th} are constants. The partial derivative $\frac{\partial \mathcal{L}_{CE}}{\partial S_i}$ determines the contribution of spike S_i to the overall loss gradient.

if x_i deviates significantly from c, $\operatorname{sech}^2(\cdot)$ would approach zero, causing the overall gradient to vanish during backpropagation. Conversely, if x_i is extremely close to c and kc is extremely small, the gradient may become excessively large:

$$tanh(kc) \approx kc \text{ if } kc \ll 1 \text{ , and } x \to c$$

$$\implies g\prime(x_i) \approx \frac{k \cdot \operatorname{sech}^2(0)}{2kc} = \frac{1}{2c}$$

Gradients may explode as $c \to 0$. Therefore, it is important to keep c and k within appropriate ranges.

We demonstrate the effectiveness of AMP2 in the context of MP initialization. Since x_i depends on MP_i^1 (**Equation 8**), we compare AMP2 with two other prevailing strategies for MP_i^0:

(1) Zero initialization (Pure Input: $MP_i^0 = 0$)
This commonly used approach completely ignores residual MP between spatially adjacent spiking neurons. Compensation is typically achieved through timestep-wise accumulation. Since X_i is produced by a normalization layer, its expectation tends to remain constant and nonadjustable.

$$\mathbb{E}[x_i] = \mathbb{E}[\frac{\beta MP_i^0 + X_i}{V_{th}}] = \frac{\mathbb{E}[X_i]}{V_{th}} \approx 0$$

(2) Random initialization (Pure Perturbation: $\alpha = 0$)
Ren et al. (2023a) proposed this method to enhance generalization by introducing perturbations to MP. While such perturbations do benefit Spiking PointNet, the method also relies on increasing timesteps to maintain this effect. The updating efficiency of all gradients is adjusted by β.

$$\mathbb{E}[x_i] = \frac{\beta \mathbb{E}[MP_{random}] + \mathbb{E}[X_i]}{V_{th}} \approx \frac{\beta c}{2V_{th}}$$

(3) AMP2 (Memory-Perturbation Fusion)
From a neuroscience perspective, spatially proximate neurons can influence each other's states through molecular diffusion, even without direct synaptic connections. This phenomenon can be observed in AMP2 that MP_i^1 would

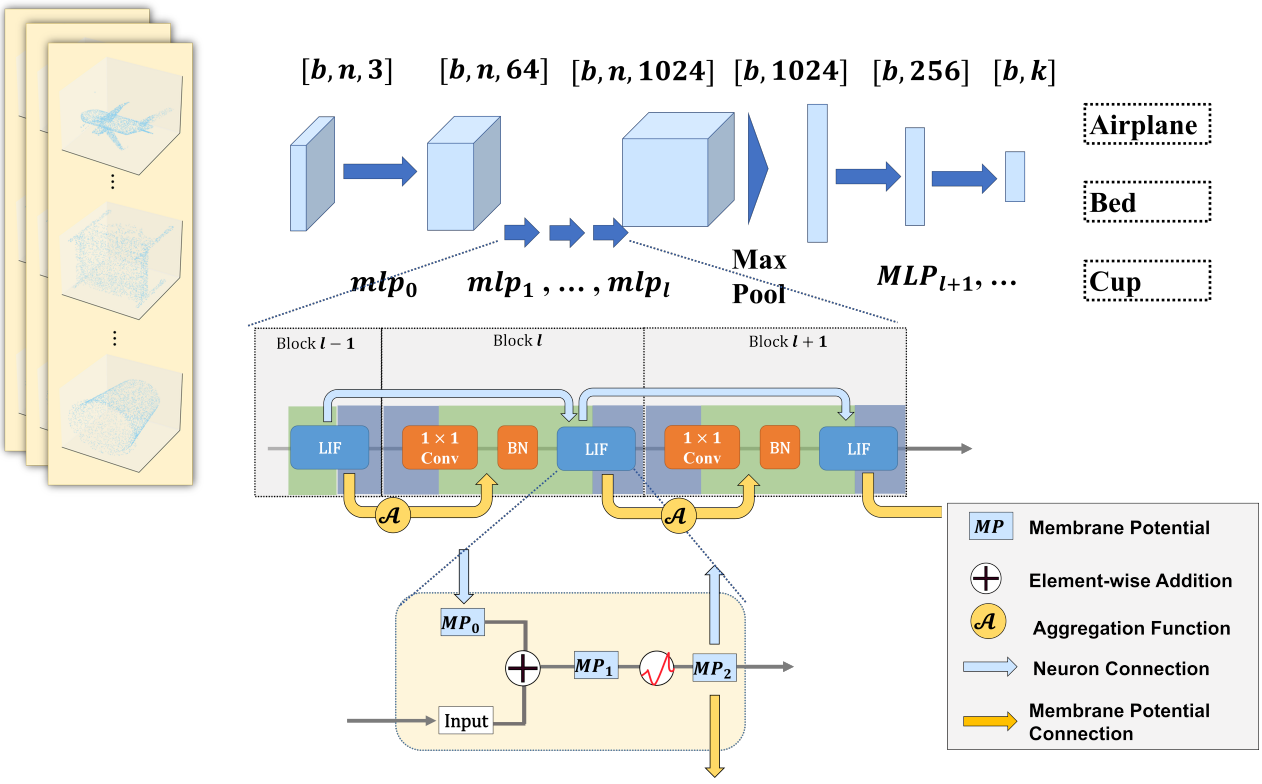

Figure 2: The workflow of Activation-wise Membrane Potential Propagation (AMP2) applied in Spiking PointNet.

	Type	Targets
A	Vanilla	$f(Spike, Features)$
B	Spike-Element-Wise	$f(Spike, Spike)$
C	Membrane	$f(Features, Features)$
D	AMP2	$f(MP, Features)$

Table 2: Definition of three existing residual connections (A-C) in SNNs and AMP2 (D).

integrate input X_{i-k} from adjacent neurons. Rather than random policy aiming for uniform gradient updates across depth, AMP2 encourages depth-related gradient updates. Furthermore, although we replace timestep-wise accumulation with spatially adjacent accumulation to save computational resources, AMP2 remains fundamentally compatible with both approaches. This can be interpreted as a global neuron leveraging an ensemble effect.

$$\mathbb{E}[MP_i^1] \approx \beta(1-\alpha)\frac{c}{2}\cdot\frac{1-(\beta\alpha)^i}{1-\beta\alpha}$$

$$\lim_{i\to\infty}\mathbb{E}[x_i] = \lim_{i\to\infty}\frac{\mathbb{E}[MP_i^1]}{V_{th}} = \frac{\beta c(1-\alpha)}{2V_{th}(1-\beta\alpha)}$$

Residual MP Connection (RMP)

We compare three existing residual shortcuts (A, B, C) with the one used in AMP2 (D) in **Table 2** and **Figure 3**. Zheng et al. (2021) introduced a vanilla shortcut between spikes

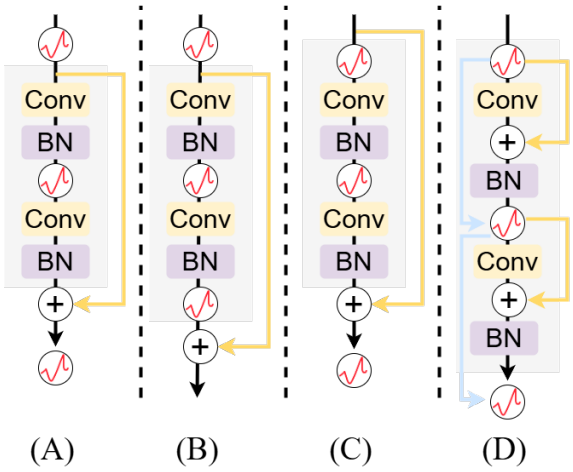

Figure 3: Illustration of existing shortcuts (A-C) in SNNs and AMP2 (D). Yellow lines indicate residual connection paths, while blue line denotes activation-wise propagation.

and normalized features, mimicking the original residual connection. While this preserves the spike-driven nature, it lacks the identity mapping function. Fang et al. (2021) presented an alternative path for pure spikes; this SEW approach transforms spike-driven characteristics into integer-driven ones. Subsequently, Hu et al. (2024) exploited the

Activation	AMP2	Space	Time
ReLU	✗	$O(BCN)$	$O(BCN)$
LIF	✗	$O(2TBCN)$	$O(TBCN)$
	✓	$O(2BCN)$	$O(BCN)$

Table 3: Complexity comparison of ReLU, timestep-based LIF and activation-based LIF

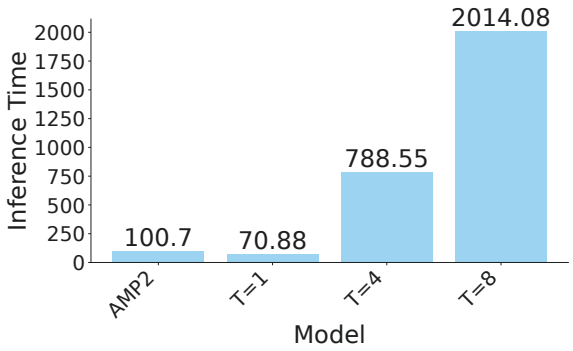

Figure 4: Impact of timesteps $(1, 4, 8)$ on the inference time (seconds) of Spiking PointNet for a single batch, evaluated on the neuromorphic hardware Lynxi HP201.

Membrane Shortcut to address previous limitations. Although this shortcut is claimed to connect MP, it actually integrates normalized features that have not been accumulated as MP. In contrast, we propose a novel approach that establishes a genuine connection between membrane potentials.

Complexity and Energy Efficiency

Energy consumption in SNNs is primarily dominated by accumulate operations (AC) and multiply-accumulate operations (MAC). According to measurements by Horowitz (2014), a single 32-bit floating-point MAC consumes $4.6\,\mathrm{pJ}$ (E_{MAC}), while an AC operation consumes only $0.9\,\mathrm{pJ}$ (E_{AC}). Based on **Table 3**, the energy consumption of standard LIF processing can be estimated by **Equation 11**, whereas the consumption of AMP2 depends on the specific element-wise functions $f(\cdot)$. The power costs of ADD and AND are estimated by **Equation 12** and **Equation 13**, respectively. It is evident that AMP2 reduces the computational overhead compared to the commonly used 4 or 8 timesteps. A more intuitive comparison on inference latency can be seen in **Figure 4**.

$$E_{LIF} = TBCN \cdot E_{MAC} \tag{11}$$

$$E_{AMP2(+)} = BCN \cdot (E_{AC} + 2 \cdot E_{MAC}) \tag{12}$$

$$E_{AMP2(\times)} = 3BCN \cdot E_{MAC} \tag{13}$$

Experiment

3D Point Cloud Recognition

From **Table 5**, we can see that AMP2 enables both Spiking PointNet and Spiking PointNet++ to achieve superior performance at a single timestep on ModelNet40 and ScanObjectNN. Notably, AMP2 not only allows SNNs to outperform conventional counterparts training with 4 timesteps, but also surpasses other low-timestep approaches, such as TRR and MP Smooth+Guidance. Moreover, AMP2 facilitates lightweight SNNs in achieving marginal outperformance over significantly larger baseline models, including E3DSNN, P2SResLNet, and Spiking Point Transformer. AMP2-based Spiking PointNet++ with multi-scale grouping (MSG) demonstrates higher overall accuracy than single-scale (SSG), achieving 91.63% on ModelNet40 and 78.81% on ScanObjectNN, which represents the best results among point cloud SNNs to date.

Type	AMP2	T	SharpNetPart(↑)		S3DIS(↑)	
			Instance	Class	Point	Class
ANN	✗	1	84.48	80.61	79.59	54.38
SNN	✗	1	78.46	72.76	70.66	46.10
SNN	✗	4	81.38	75.53	72.42	**47.79**
SNN	✓	1	**82.44**	**78.31**	**74.21**	46.70

Table 4: Effect of AMP2 on Spiking PointNet for semantic segmentation task of point cloud

We assessed the effectiveness of AMP2 in improving the 3D semantic segmentation performance of Spiking PointNet on two datasets: ShapeNetPart (Yi et al. 2016) and the Stanford 3D Indoor Scene (Armeni et al. 2016) Dataset (S3DIS). As presented in **Table 4**, AMP2 enables Spiking PointNet to achieve superior mean intersection over union (mIoU) for both instance and category segmentation compared to the 4-timestep model. Additionally, on the S3DIS dataset, AMP2 surpasses the 4-timestep SNN in point recognition accuracy.

Neuromorphic Recognition

From **Table 6**, we evaluate AMP2 on neuromorphic data DVS128Gesture (Amir et al. 2017) using both frame-based and point-based inputs. For frame input, AMP2 is combined with timestep-wise updating on the ResNet-tiny architecture (7B-Net) proposed by Fang et al. (2021). Consequently, AMP2 consistently provides stable improvements for three variant SNNs with 4 timesteps.

Ablation Study

Influence of Network Depth

In **Table 7**, we conduct an ablation study on the two components of AMP2: Activation-wise Propagation (AWP) and Residual Membrane Potential Connection (RMP). Additionally, to assess the impact of network depth on AMP2, we vary the number of Set Abstraction (SA) modules in Spiking PointNet++ (SSG) to 3 (baseline), 4, and 6. We observe a decreasing trend in performance as the number of SA blocks increases, attributed to the absence of explicit residual connections in Spiking PointNet++. When equipped with either AWP or RMP alone, consistent accuracy improvements are observed on ModelNet40.

We further evaluate AMP2 on transformer-based SNN, including Spike-Driven Transformer V2 (SDT-v2) (Yao et al.

Arch.	Method	Type	T(↓)	ModelNet40(↑)		ScanObjectNN(↑)	
				OA	mACC	OA	mACC
PointNet	PointNet†	ANN	1	90.43	86.66	68.40	62.38
	Spiking PointNet†	SNN	1	87.36	82.58	24.84	26.51
		SNN	4	88.41	83.43	64.04‡	60.14‡
		SNN	8	88.50	84.73	-	-
	ANN2SNN	SNN	16	88.17	-	**66.56**	-
	TRR	SNN	2	88.84	-	-	-
	AMP2(Ours)	SNN	1	**89.74**	**85.60**	65.48	**60.29**
PointNet++(SSG)	PointNet++†	ANN	1	92.33	90.07	84.46	82.62
	Spiking PointNet++†	SNN	1	87.36	82.58	48.01	43.77
		SNN	4	91.22	87.43	-	-
		SNN	8	91.61	88.59	-	-
	ANN2SNN	SNN	16	89.45	-	69.22	-
	MP Smooth+Guidance	SNN	2	91.13	-	-	-
	TRR	SNN	1	89.65	-	-	-
	TRR	SNN	2	90.57	-	-	-
	AMP2(Ours)	SNN	1	**91.23**	**88.99**	72.57	67.78
PointNet++(MSG)	AMP2(Ours)	SNN	1	**91.63**	**89.16**	**78.81**	75.71
Efficient 3D SNN	E3DSNN	SNN	1	91.50	-	-	-
KPConv	P2SResLNet-B	SNN	1	90.60	89.20	74.46‡	72.58‡
	KPConv-SNN	SNN	40	70.5	67.6	43.90	38.70
Point Transformer	SPT(Q-SDE768)	SNN	1	90.87	-	76.33	-
		SNN	2	91.13	88.93	77.03	-
		SNN	4	91.22	88.45	78.03	**75.87**

Table 5: Classification accuracy of existing SNNs at low timesteps on ModelNet40 and ScanObjectNN. The best results for each metric are highlighted in **bold**. † indicates self-reproduced results, while ‡ denotes results reproduced by Wu et al. (2025). T indicates timesteps used in SNNs.

Input	Arch.	Method	Type	AMP2	T(↓)	Acc.
Frame	VGG-9	SSNN	SNN	✗	5	90.74
		TRR				91.67
		SLT				89.35
		MP Smooth+Guidance				93.23
	ResNet-tiny	SEWResNet†	SNN	✗	4	88.19
				✓		**93.75(+5.56)**
		Spiking ResNet†		✗		71.88
				✓		**82.29(+10.41)**
		Spiking PlainNet†		✗		75.0
				✓		**78.13(+3.13)**
	Transformer	SpikingResformer-Ti	SNN	✗	5	90.63
		Spikformer			5	79.52
		SDT-v1(2-256)			5	92.24
		SDT-v1(2-256)			16	99.30
Point	PointNet	PointNet†	ANN	✗	1	89.57
		Space-time Event Cloud	ANN	✗	1	88.77
		Spiking PointNet†	SNN	✗	1	31.22
			SNN	✗	4	91.32
		AMP2(Ours)	SNN	✓	1	**92.82(+61.6)**
	PointNet++	PointNet++†	ANN	✗	1	96.74
		Space-time Event	ANN	✗	5	95.70
		Spiking PointNet++†	SNN	✗	1	85.0
			SNN	✗	4	94.68
		AMP2(Ours)	SNN	✓	1	**92.89(+7.89)**
	SpikePoint	SpikePoint	SNN	✗	16	**98.74**

Table 6: Comparative results (%) of existing low-timesteps SNNs on DVS128Gesture. † represents self-reproduced experiments.

SA	Param(M)	AMP2 AWP	AMP2 RMP	Acc.(%)
3	1.4757	✗	✗	87.35
		✗	✓	89.88
		✓	✗	88.55
		✓	✓	91.23
4	1.4833	✗	✗	83.58
		✗	✓	89.17
		✓	✗	85.07
		✓	✓	90.30
6	1.8169	✗	✗	78.71
		✗	✓	89.29
		✓	✗	80.46
		✓	✓	89.65

Table 7: Impact of the Set Abstraction (SA) module and AMP2 components on the classification accuracy of Spiking PointNet++ (SSG) for ModelNet40. AWP: Activation-wise Propagation; RMP: Residual Membrane Potential connection.

Model	Param(M)	T	AMP2 AWP	AMP2 RMP	Acc.(%)
SDT-v2	15	1	✗	✗	80.91
			✗	✓	N/A
			✓	✗	80.08
			✓	✓	80.96
Spikformer	29.68	1	✗	✗	78.0
		4	✗	✗	80.36

Table 8: Performance of transformer-based SNN on ImageNet100. SDT-v2 refers to Spike-Driven Transformer V2 (8-256), and Spikformer denotes the 8-512 variant. T indicates timesteps.

2024a) and Spikformer (Zhou et al. 2023) with default surrogates on an 100-category subset of ImageNet. Results in **Table 8** confirm the effectiveness of AMP2 in enhancing performance. Particularly, removing AWP while keeping only RMP causes extremely slow loss convergence on SDT-v2, indicating the transformer architecture's reliance on AWP.

Training Curves

Figure 5 demonstrates the influence of AMP2 and its components AWP and RMP on the training of Spiking PointNet++ (SSG). Compared with the baseline trained with a single timestep, AWP yields consistent improvements across nearly all epochs. Similarly, the complete AMP2 configuration achieves faster accuracy gains than RMP during the first 50 epochs.

Firing Rate

Apart from accuracy, firing rate serves as a key indicator of energy efficiency in SNNs. As shown in **Table 9**, we report the spike firing rate of all activation layers within Spiking PointNet++. Each SA module contains three spiking activation layers, we compute the average firing rate across them

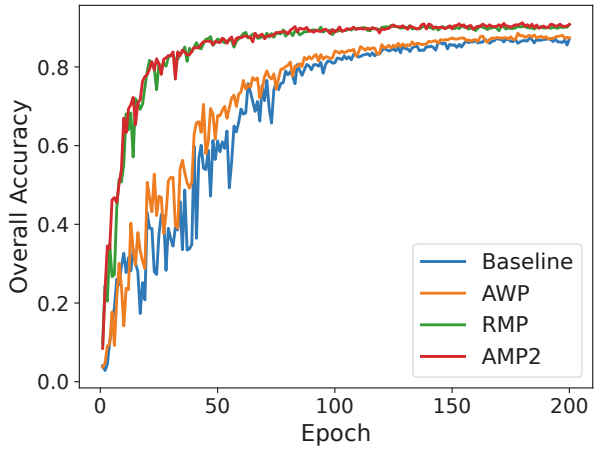

Figure 5: Training curves of Spiking PointNet++ (SSG) with varying configuration on ModelNet40.

Spiking PointNet++	Firing Rate (↓) w/o AMP2	w/ AMP2
Cls. LIF1	**23.78**	26.11
Cls. LIF2	**33.78**	36.10
SA1 LIF Avg.	31.44	**24.70**
SA2 LIF Avg.	27.23	**22.35**
SA3 LIF Avg.	22.29	**18.23**

Table 9: Firing rate of Spiking PointNet++ (SSG) in each spiking activation layer.

to represent the module's overall spiking activity. A consistent decline in firing rate across the three SA modules suggests progressively more efficient feature extraction. In contrast, the classification head exhibits a higher firing rate, indicating increased energy use for recognition—an intentional design that prioritizes task-critical computation while reducing overall overhead.

Conclusion And Discussion

In this paper, we propose **A**ctivation-wise **M**embrane **P**otential **P**ropagtion (AMP2) to enable single-timestep training and inference for SNNs. AMP2 is comprised of Activation/Neuron-wise Propagation (AWP) and Residual Membrane Potential Connection (RMP). AWP is a brain-inspired strategy that replaces timestep-wise membrane potential accumulation with accumulation across spatially adjacent neurons. RMP introduces a genuine fusion for membrane potential in SNN.

Future work. This study leaves two directions for future exploration. First, given sufficient computational resources, it is worth evaluating AMP2 on large-scale datasets such as ImageNet-1k and ES-ImageNet. Second, collaboration with hardware team is encouraged, as current neuromorphic hardware still lacks basic capabilities, like precise power measurement for SNN. We hope AMP2 can bridge the gap between efficient SNN and real-world applications.

Acknowledgments

This work was supported by the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and NSFC General Program (Grant No. 62176215 and Grant No. 62573362)

References

Ambityga. 2023. ImageNet100. https://www.kaggle.com/datasets/ambityga/imagenet100.

Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; Kusnitz, J.; Debole, M.; Esser, S.; Delbruck, T.; Flickner, M.; and Modha, D. 2017. A Low Power, Fully Event-Based Gesture Recognition System. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7388–7397.

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1534–1543.

Ding, Y.; Zuo, L.; Jing, M.; He, P.; and Deng, H. 2025. Rethinking spiking neural networks from an ensemble learning perspective. *arXiv preprint arXiv:2502.14218*.

Ding, Y.; Zuo, L.; Jing, M.; He, P.; and Xiao, Y. 2024. Shrinking your timestep: Towards low-latency neuromorphic object recognition with spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11811–11819.

Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; and Tian, Y. 2021. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34: 21056–21069.

Horowitz, M. 2014. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*, 10–14. IEEE.

Hu, Y.; Deng, L.; Wu, Y.; Yao, M.; and Li, G. 2024. Advancing Spiking Neural Networks Toward Deep Residual Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Ren, D.; Ma, Z.; Chen, Y.; Peng, W.; Liu, X.; Zhang, Y.; and Guo, Y. 2023a. Spiking pointnet: Spiking neural networks for point clouds. *Advances in Neural Information Processing Systems*, 36: 41797–41808.

Ren, H.; Zhou, Y.; Huang, Y.; Fu, H.; Lin, X.; Song, J.; and Cheng, B. 2023b. Spikepoint: An efficient point-based spiking neural network for event cameras action recognition. *arXiv preprint arXiv:2310.07189*.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.

Wang, Q.; Zhang, Y.; Yuan, J.; and Lu, Y. 2019. Space-time event clouds for gesture recognition: From RGB cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1826–1835. IEEE.

Wu, P.; Chai, B.; Li, H.; Zheng, M.; Peng, Y.; Wang, Z.; Nie, X.; Zhang, Y.; and Sun, X. 2025. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 21563–21571.

Wu, Q.; Zhang, Q.; Tan, C.; Zhou, Y.; and Sun, C. 2024. Point-to-Spike Residual Learning for Energy-Efficient 3D Point Cloud Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6092–6099.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Yao, M.; Hu, J.; Hu, T.; Xu, Y.; Zhou, Z.; Tian, Y.; XU, B.; and Li, G. 2024a. Spike-driven Transformer V2: Meta Spiking Neural Network Architecture Inspiring the Design of Next-generation Neuromorphic Chips. In *The Twelfth International Conference on Learning Representations*.

Yao, M.; Hu, J.; Zhou, Z.; Yuan, L.; Tian, Y.; XU, B.; and Li, G. 2023. Spike-driven Transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yao, M.; Qiu, X.; Hu, T.; Hu, J.; Chou, Y.; Tian, K.; Liao, J.; Leng, L.; Xu, B.; and Li, G. 2024b. Scaling Spike-driven Transformer with Efficient Spike Firing Approximation Training. *arXiv preprint arXiv:2411.16061*.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3D shape collections. *ACM Trans. Graph.*, 35(6).

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16259–16268.

Zheng, H.; Wu, Y.; Deng, L.; Hu, Y.; and Li, G. 2021. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11062–11070.

Zhou, C.; Zhang, H.; Zhou, Z.; Yu, L.; Huang, L.; Fan, X.; Yuan, L.; Ma, Z.; Zhou, H.; and Tian, Y. 2024a. Qkformer: Hierarchical spiking transformer using qk attention. *Advances in Neural Information Processing Systems*, 37: 13074–13098.

Zhou, Z.; Che, K.; Fang, W.; Tian, K.; Zhu, Y.; Yan, S.; Tian, Y.; and Yuan, L. 2024b. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket. *arXiv preprint arXiv:2401.02020*.

Zhou, Z.; Zhu, Y.; He, C.; Wang, Y.; YAN, S.; Tian, Y.; and Yuan, L. 2023. Spikformer: When Spiking Neural Network Meets Transformer. In *The Eleventh International Conference on Learning Representations*.

Zuo, L.; Ding, Y.; Luo, W.; Jing, M.; Tian, X.; and Yang, K. 2024. Temporal reversed training for spiking neural networks with generalized spatio-temporal representation. *arXiv e-prints*, arXiv–2408.